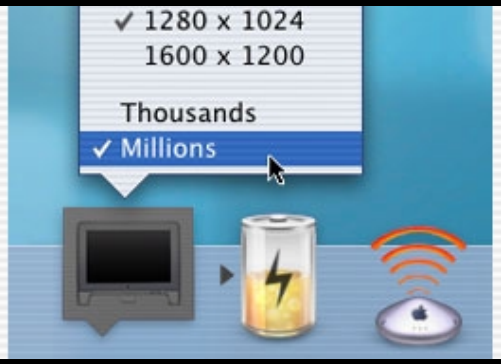
Session 310

AFP 3.0 and AFP Client for Mac OS X



Leland Wallace
Senior Engineer, iServers

Introduction

- Go over AFP 3.0 as implemented in Mac OS X
- Talk about the AFP Client in Mac OS X
- Go over the User Authentication Module (UAM) SDK

AFP 3.0 on Mac OS X

- History and goals
- What's new
- Current implementation

History and Goals

- AFP 3.0 is analogous to HFS+
- Needed to support a similar API over the network
- Allow for a more Unix like view of the file system
  - File level permissions
- Support large file sizes and long filenames
- Support mixed language names

What's New In AFP 3.0

- AFP 3.0 is a superset of all prior versions
  - We did remove PRODOS support
- Long Unicode names
- 64 bit file sizes
- Optional Unix permissions
- Support for long attentions
- Support for directory services
- Support for reconnect
- GMT dates

New Datatypes

- AFPName
  - 1 byte path type (0x03)
  - 4 byte encoding hint (Filesystem objects only)
  - 2 byte data length
  - Up to 765 bytes of UTF-8 data

Current Implementation

- Not currently implemented in Mac OS X
  - Reconnect
  - Long attentions
  - Dir services support
- The version string is “AFPX03”

AFP 3.0 Calls Overview

- New calls
- FPByteRangeLockExt (0x3B)
- FPReadExt (0x3C)
- FPWriteExt (0x3D)
- FPGetAuthMethods (0x3E)
- FPLoginExt (0x3F)
- FPGetSessionToken (0x40)
- FPDisconnectOldSession (0x41)

AFP 3.0 Calls Overview

- Changed calls
- FPGetServerInfo (0x0F)
  - New flag bits, directory names
- FPGetSrvrParms (0x10)
  - UTF-8 volume names for AFP 3.0
- FPMap ID (0x1F)
  - Support for UTF-8 names
- FPMapName (0x20)
  - Support for UTF-8 names

AFP Client Overview

- Goals
  - Provide equivalent functionality to the Mac OS 9 client
- Last years talk
  - Minimal API changes

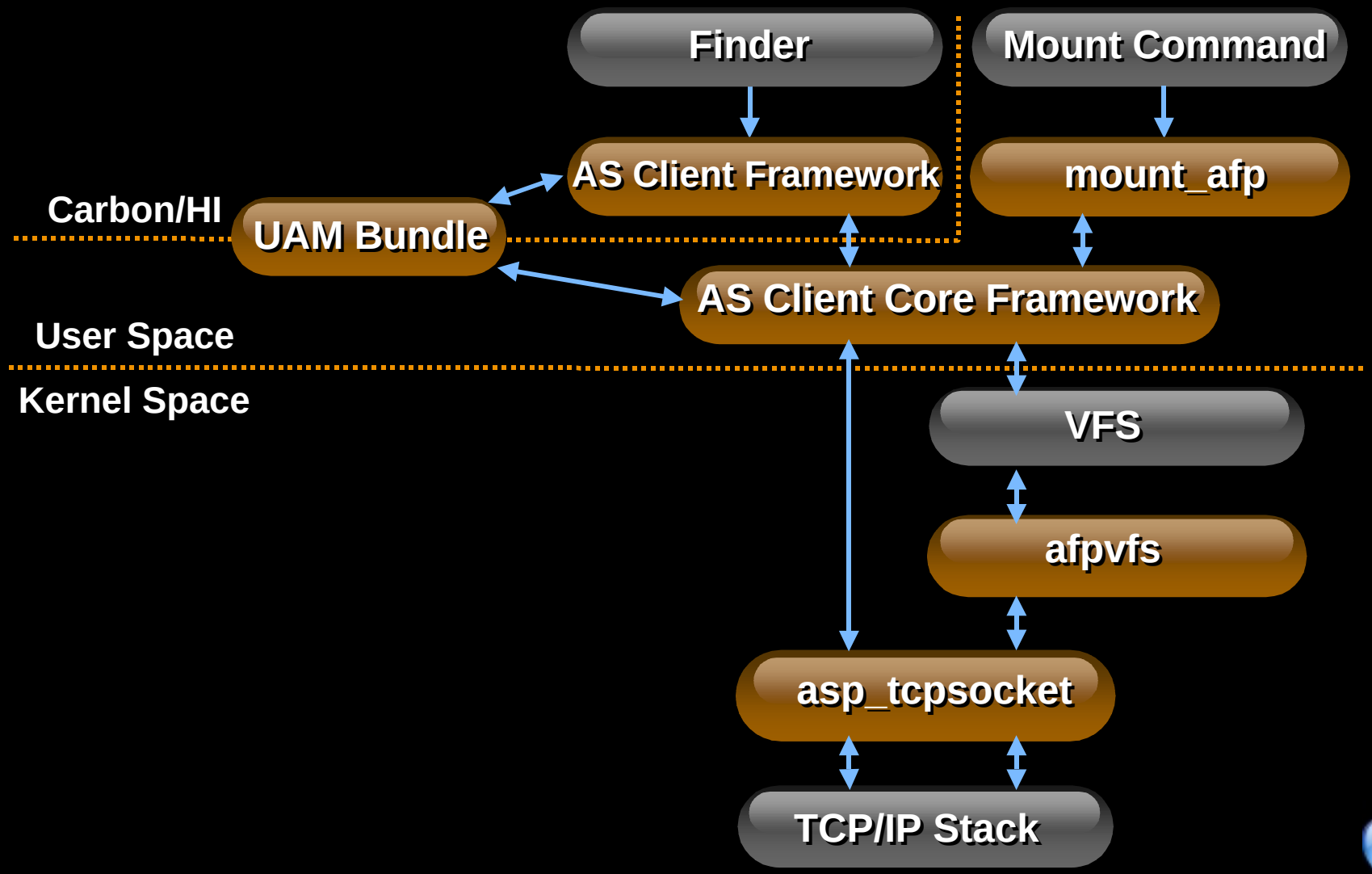
AFP Client: User Experience

- Finder
- Classic
- Alias resolution
- Login window
- Keychain
- Automounter
- International

AFP Client: Developer Experience

- OS Layering
- APIs
  - Shared Volume Enumerator
  - AFP URLs
  - User Authentication Modules

AFP Client Architecture



Shared Volume Enumerator API

- New calls to support Mac OS X
  - AFPCreateSVEFromAddress()
  - AFPMountSharedVolumeOnMP()
  - AFPChangePassword()
- Requires Carbon
  - Link with CarbonCore if you use this API

AFP URLs

- afp://[user:password@]server[:port]/volume
- afp:/at/[user:password@]server[:zone]/volume
- For example:
  - afp://Leland:foo@myserver.com/myVolume

AFP URLs

- Can also specify the UAM by inserting
  - user[;AUTH=uamname]:password
- For example:
  - **afp://;AUTH=No%20User%20Authent@server.com/myVolume**

AFP URL API

- NewAFPURL()
- DisposeAFPURL()
- IsAFPURL()
- ParseAFPURL()
- MountAFPURL()

User Authentication Modules

- Overview
- Changes from Mac OS 9.x
- Bundle structure
- Entrypoint
- Callbacks
- Porting Mac OS 9.x UAMs
- Example

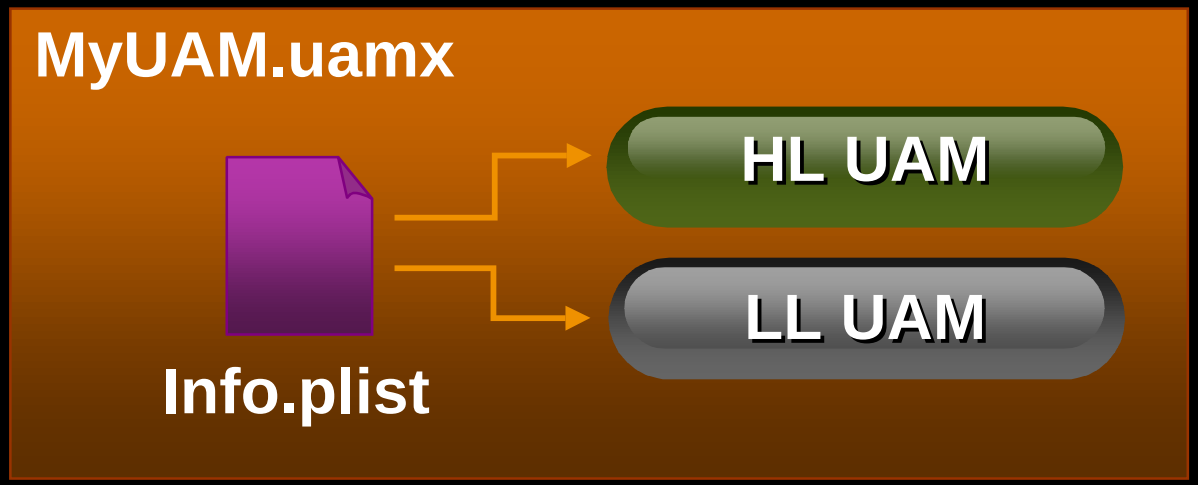
Client UAM Overview

- Bundles of type ‘uamx’
- Located in /Library/Filesystems/AppleShare/Authentication
- Loaded with CFBundle calls
- Not a CFPlugin
- Contains high level and low level sub-bundles
- May be loaded without a user context

Changes from Mac OS 9.x UAM API

- UAM is passed a struct sockaddr* instead of an OTAddress*
- There are additional fields in the UAMArgs structure for UTF-8 usernames and userDomains
- sessionRefnum is now a long (was a short)
- Async callbacks not supported

UAM Bundle Structure



- High Level UAM (links with Carbon)
- Low Level UAM
- Info.plist

UAM Info.plist Keys

- UAMProtocolName <String>
- UAMMaxPasswordLength <Number>
- UAMHighLevelEntrypoint <Dictionary>
  - UAMBundleIdentifier <String>
  - UAMBundlePath <String>
  - UAMEntrypoint <String>
- UAMLowLevelEntrypoint <Dictionary>
- UAMUserVisibleName <String> (localized)

UAM: Entrypoint

- Function name specified in Info.plist
- OSStatus UAMEntryPoint(UAMArgs* args)

UAM: Entrypoint

- Command selectors:
  - kUAMOpen
  - kUAMLogin
  - kUAMClose
- Optional selectors:
  - kUAMPWDlog / kUAMVSDlog
  - kUAMGetInfoSize / kUAMGetInfo
  - kUAMChgPassDlog / kUAMChgPass

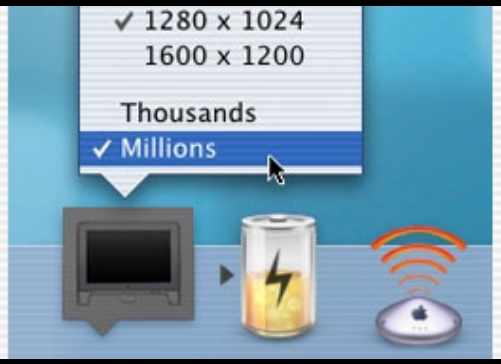
UAM: Callbacks

- GetClientInfo()
- OpenSession()
- SendRequest()
- CloseSession()

Porting a Mac OS 9 UAM

- Compile your main code and resources into the High Level Bundle (you need to be Carbon compliant)
- Fill out the main bundles UAMHighLevelEntrypoint Info.plist key
- Put your icon into the main bundle
- Support the new features of the Mac OS X UAM API at your leisure

Example UAM



Further Information

- AFP 3.0 documentation is at
  http://developer.apple.com/techpubs/macosx/Networking/networking.html
- A Beta UAM SDK is available at
  http://developer.apple.com/sdk/
- The Directory Services Documentation is at
  http://developer.apple.com/techpubs/macosx/Networking/networking.html
- The other AFP Client APIs are in
  - AppleShareClient.framework
  - AppleShareClientCore.framework

Roadmap

301 Mac OS X Server
The Server side of the equation

Room A1
Tues., 2:00pm

302 Directory Services
Server-side authentication

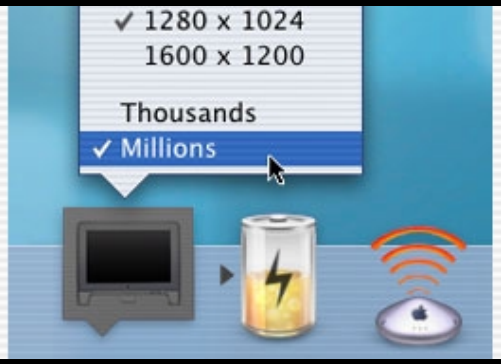
Room A1
Tues., 3:30pm

Who to Contact

Tom Weyer
Network & Communications Technology Manager
Apple Worldwide Developer Relations
weyer@apple.com

http://developer.apple.com/events/urls.html

Q&A



Tom Weyer
Network & Communications Technology Manager
weyer@apple.com

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001